

```
*****
* This program will calculate the recodes for the publication tables.1999 NATIONAL FINAL
* 1/8/01
* Program:    tblrcd.sas
* Programmed by: Diane Schwartz
* Date:      June 19,1998
*
* Updated: August 15, 2000 -- PUBSUB - changed code for pubsub=8
*
* Input file:  /natdata/production/
*
* Output file: /natadata/production/
*
* Variables that are calculated are:
* Section One -- persons rostered information
*   * zinc -- income of ref. person and related hh members
*   * salinc -- salary income of ref.person and related hh members
*   * zinc2 -- total household income
*   * zadult -- number of adults 18 and over
*   * hhldkid -- number of householders children ( any age)
*   * hhkdu18 -- number of houeholders children under 18
*   * skdu18 -- number of single kids under 18
*   * skidu6 -- number of single kids under 6
*   * skd617 -- number of single kids ages 6 to 17
*   * adltdkd29 -- number of adult householders children,
*                 ages 18 to 29 who are not married
*   * adltdkd30 -- number of adult householders children,
*                 ages 30 or more who are not married
*   * kid013 -- number of children in household between ages 0 and 13
*   * kid515 -- number of children in household between ages 5 and 15
*   * kid -- number of children in household
*   * kidu6 -- number of children under 6 year of age and single
*   * kid617 -- number of children ages 6 to 17 who are single
*   * elder -- number of elderly persons ages 65 and older
*   * hhage -- householders age
*   * hhsex -- householders sex
*   * hhspan -- householders spanish origin
*   * hhrace -- householders race
*   * hhmove -- householders move in date
*   * hhmar -- householders marital status
*   * hhgrad -- householders educational level
*   * hhrel -- householders relationship code
*   * spouse -- spouse of householder is present
*   * nonrel -- number of nonrelatives in household
*   * coown -- coowner
*   * nrldgr -- non-related lodger
```

- * unrkid -- unrelated kid ages 0 to 17
- * unrelper -- unrelated person 18 and over
- * parent -- parent of householder present
- * grandpa -- grandparent of householder present
- * grankid -- grandchild of householder present
- * refper -- reference person flag
- * relfam -- number of related subfamilies
- * relhhage -- related subfamilies head of families age
- * reloth -- households with other types of relatives
- * othrel -- other relatives of householder
- * nrel2 -- 2 person household - not related
- * nrel3 -- 3-8 person household - not related
- * norel3 -- count of not related household members
- * scndry -- number of secondary families - not related to householder
- * respdnt -- household respondent that moved in the last year
- * hrmar -- household respondents marital status
- * movedly -- flag for householder moved within the last year
- *
- * Section Two -- main portion of recodes
- * phone -- phone
- * kitchen -- complete kitchen
- * rooms -- number of rooms
- * confeet - confeetr -- rounded at end to be confeet -- condominium and
 cooperative fee
- * zcrowf - zcrowfr -- rounded at end to be zcrowf -- square feet per
 person
- * zcrowd - zcrowdr -- rounded at end to be zcrowd -- persons per room
- * zcrowb - zcrowbr -- rounded at end to be zcrowb -- persons per bedroom
- * tprice -- purchase price
- * zaq1 -- tally for upkeep in selected physical problems
- * zaq2 -- tally for hallways in selected physical problems
- * zadeq -- tally to determine severe or moderated selected physical problems
- * maint - maintr -- rounded at end to be maint -- routine maintenance in
 last year
- * zsmhcn - zsmhcnr -- rounded at end to be zsmhcn -- monthly housing costs
 withour maintenance
- * zsmhcm - zsmhcml -- rounded at end to be zsmchm -- monthly housing costs
 with maintenance
- * zsmhco - zsmhcrl -- rounded at end to be zsmhco -- monthly housing costs
 excluding 2nd and subsequent mortgages and maintenance
- * zsmhcv - zsmhcvr -- rounded at end to be zsmhcv -- monthly housing costs
 including vacant-for rent but excluding maintenance
- * zsmhcp - zsmhcpr -- rounded at end to be zsmhcp -- monthly housing costs
 as a percent of current income
- * zvi - zvir -- rounded at end to be zvi -- ratio of value to current income
- * amtomo - amtomor -- rounded at end to be amtomo -- average monthly costs

- * for fuel oil
- * amtimo - amtimor -- rounded at end to be amtimo -- average monthly costs
- * for property insurance
- * amtwmo - amtwmor -- rounded at end to be amtwmo -- average monthly costs
- * for water
- * amttmo - amttmor -- rounded at end to be amttmo -- average monthly costs
- * for trash
- * amtgmo - amtgmor -- rounded at end to be amtgmo -- average monthly costs
- * for bottled gas
- * amtfmo - amtfmor -- rounded at end to be amtfmo -- average monthly costs
- * for other fuel
- * lrntmo - lrntmor -- rounded at end to be lrntmo -- land rent fee paid
- * amtxp - amtxpr -- rounded at end to be amtxp -- average monthly costs
- * for real estate taxes
- * amtxv - amtxvr -- rounded at end to be amtxv -- annual taxes paid per
\$1000 value
- * newc -- new construction
- * pubsub -- rent reductions
- * saving -- savings
- *
- * Section Three -- poverty lookup table
- * poor - poorr -- rounded at end to be poor -- household income as percent
of poverty level
- * pvlkup -- not used in tables, only for verification -- key to access the
poverty lookup table
- * povlvl -- not used in tables, only for verification -- value in poverty
lookup table corresponding to lookup key
- *
- * Section Four -- calculations for INCM03 stub
- * salinc -- wages and salary
- * inc3flg -- flag determining if 2 or more people each earned over 20% of
wages and salary
- * per20wg -- not used in tables, only for verification -- 20% wage level
- *
- * Section Five -- first part of the mortgage recodes
- * otpin - otpinr -- rounded at end to be otpin -- outstanding principle and
interest
- * clpeva - clpevar -- rounded at end to be clpeva -- current load as percent
of value
- * crintr - crinrr -- rounded at end to be crintr -- current interest rate
- *
- * Section Six -- last part of the mortgage recodes
- * rymort -- remaining years mortgaged
- * prin01 - prin01r -- rounded at end to be prin01 -- monthly payment for
principle and interest
- * paypm1 -- payment plan for primary mortgage

- * paypm2 -- payment plan for secondary mortgage
- * credlp -- type of home equity mortgages
- *
- * Section Seven -- recent mover recode
- * amvone -- household all moved from one unit
- * mv1hh -- householder of previous unit status of moving from one unit
- * mv2hh -- householder of previous unit status of moving from 2+ units
- * amvtwo -- household moved from 2+ units
- * smv3 -- some already here, rest moved in
- * mv3hh -- householder of previous unit status of moving when some were here
- * mv4nr -- previous units not reported
- * lastyr -- total with a move in the last year
- * xheadhh -- householders xhead value
- * xunithh -- householders xunit value
- * xcosthh -- householders xcost value
- * xperhh -- householders xper value
- * xinushh -- householders xinus value
- * xtenhh -- householders xten value
- * xrelhh -- householders xrel value
- * rmvunvh -- recent movers universe for the householder
- *
- * Section Eight -- journey to work recodes
- * j1drv -- drives self
- * j1cp -- carpool
- * j1cp2 -- carpool - 2 person
- * j1cp3 -- carpool - 3 person
- * j1cp4 -- carpool - 4+ person
- * j1mass -- mass transportation
- * j1taxi -- taxicab
- * j1bike -- bicycle or motorcycle
- * j1walk -- walks only
- * j1oth -- other meansa
- * j1home -- works at home
- * j2u15 -- under 15 minutes
- * j21529 -- 15 to 29 minutes
- * j23044 -- 30 to 44 minutes
- * j24559 -- 45 to 59 minutes
- * j26089 -- 60 to 89 minutes
- * j2o90 -- over 90 minutes
- * j2home -- works at home
- * j2noplc -- no fixed place of work
- * j3u1 -- under 1 mile
- * j314 -- 1 to 4 miles
- * j359 -- 5 to 9 miles
- * j31019 -- 10 to 19 miles
- * j32029 -- 20 to 29 miles

- * j33049 -- 30 to 49 miles
- * j3o50 -- over 50 miles
- * j3home -- works at home
- * j3noplc -- no fixed place of work
- * j4123a -- 12 to 3 am
- * j436a -- 3 to 6 am
- * j467a -- 6 to 7 am
- * j47730a -- 7 to 7:30 am
- * j47308a -- 7:30 to 8 am
- * j48830a -- 8 to 8:30 am
- * j48309a -- 8:30 to 9 am
- * j4910a -- 9 to 10 am
- * j4104p -- 10 am to 4 pm
- * j4412p -- 4 to 12 pm
- * j4nr -- departure time not reported
- * hhwlineq -- householders wlineq value
- * hhtran -- householders tran value
- * hhvehcl -- householders vehcl value
- * hppass -- householders pass value
- * hhhjob -- householders job value
- * hhtimej -- householders timej value
- * hhdistj -- householders distj value
- * hhampm -- householders ampm value
- * hhwtime -- householders wtime value
- * jtwtot -- total persons that worked last week
- *
- * Section Nine -- RENT01 - lodgers recode and spouses grandparents recodes
- * lodg - lodgr -- rounded at end to be lodg -- rent paid by lodgers
- * ldgrs -- flag indicating it was a lodger
- * sgrndpa -- spouses grandparent found -- used in calculating gen3 for COMP05
- * sprntln -- spoused parents line number
- *
- * Section Ten -- Combine previous data sets and do all rounding
- * confeet
- * zcrowf
- * zcrowd
- * zcrowb
- * zsmhcн
- * zsmhcм
- * zsmhcо
- * zsmhcв
- * zsmhcр
- * zvi
- * amtomo
- * amtimo
- * amtwmo

```

* amttmo
* amtgmo
* amtfmo
* lrntmo
* amtxp
* amtxv
* maint
* poor
* otpin
* clpeva
* crintr
* prin01
* lodg
*
* Section Eleven -- final section -- resets missing values in rounded
* variables and calculates gen3
* gen3 -- three generations live in household -- COMP05
*
*****/*****
libname edt "$EDTS";
libname bridg "$BRIDGE";
libname rcd "$RECD";
libname temp "$TEMP";

missing D R B X M N U _ ;
/* *****/
/*      S E C T I O N   O N E           */
/* *****/
/* ** this section creates a data set with the recodes included      */
/* from the persons data set          ** */
/* *****/
data rcd.perrcd (keep = ctrlnm13 zadult zinc zinc2 elder kid013 kid515
                  hhage salinc skdu18 kidu6 kid617 adltd29 adltd30
                  spouse reloth coown rlfam relhhage
                  nrldgr unrkid unrlper kid nonrel grandpa parent
                  refper grankid othrel
                  hhsex hhrace hhspan scndry skidu6 skd617
                  hhmar nrel2 nrel3 hhgrad hhmove hhrel hhldkid respdnt
                  hrmar movedly hhkdu18 status curper datem);

length ctrlnm13 $13 zadult 3. zinc 5. zinc2 5. elder 3.
      kidu6 3. kid617 3. adltd29 3. adltd30 3.
      skdu18 3. hhage 3. kid013 3. kid515 3. salinc 5.
      spouse $1 rlfam 3. reloth 3. relhhage 3. othrel 3.
      coown 3. nrldgr 3. unrkid 3. unrlper 3. kid 3. nonrel 3.
      grandpa $1 parent $1 refper $1

```

```
grankid $1 hhsex $1 hhrace $1  
hhspan $1 scndry 3. skidu6 3. skd617 3. hhmar $1  
nrel2 $1 nrel3 $1 norel3 3. hhgrad 3. hhmove 4. hhrel 3.  
hhldkid 3. respdnt $1 hrmar $1 movedly $1 hhkdu18 3.;
```

```
merge edt.person  
edt.houshld (keep=ctrlnm13 vother status curper  
datey datem dline1) end=send;  
by ctrlnm13;
```

```
retain incnt 0;  
retain ctrlno 0;  
retain ctllst 0;  
retain outent 0;  
retain zinc 0;  
retain salinc 0;  
retain zinc2 0;  
retain zadult 0;  
retain hhldkid 0;  
retain hhkdu18 0;  
retain skdu18 0;  
retain skidu6 0;  
retain skd617 0;  
retain adltkd29 0;  
retain adltkd30 0;  
retain kid013 0;  
retain kid515 0;  
retain kid 0;  
retain kidu6 0;  
retain kid617 0;  
retain elder 0;  
retain hhage 0;  
retain hhsex '';  
retain hhspan '';  
retain hhrace '';  
retain hhmove 0;  
retain hhmar '';  
retain hhgrad 0;  
retain hhrel 0;  
retain spouse '';  
retain nonrel 0;  
retain coown 0;  
retain nrldgr 0;  
retain unrlikid 0;  
retain unrlnper 0;  
retain parent '';
```

```

retain grandpa '';
retain grankid '';
retain refper '';
retain relfam 0;
retain relhhage 0;
retain reloth 0;
retain othrel 0;
retain nrel2 '';
retain nrel3 0;
retain nrel3 '';
retain scndry 0;
retain respdnt '';
retain hrmar '';
retain movedly '';

incnt+1;
/* if(mod(incnt,500) = 0) then put 'processing obs number ' incnt; */

if last.ctrlnm13 then ctllst+1;

if first.ctrlnm13 then ctrlno+1;

if first.ctrlnm13 then
do;
  zinc = .;
  salinc = .;
  zinc2 = .;
  zadult = .;
  hhldkid = .;
  hhkdu18 = .;
  skdu18 = .;
  skidu6 = .;
  skd617 = .;
  adltdkd29 = .;
  adltdkd30 = .;
  kid013 = .;
  kid515 = .;
  kid = .;
  kidu6 = .;
  kid617 = .;
  elder = .;
  hhage = .;
  hhsex = '';
  hhspan = '';
  hhrace = '';
  hhmove = .;

```

```

hhmar = '';
hhgrad = .;
hhrel = .;
spouse = '';
nonrel = .;
coown = .;
nrldgr = .;
unrlkid = .;
unrlper = .;
parent = '';
grandpa = '';
grankid = '';
refper = '';
relfam = .;
relhhage = .;
reloth = .;
othrel = .;
nrel2 = '';
nrel3 = .;
nrel3 = '';
scndry = .;
respndnt = '';
hrmar = '';
movedly = '';

end;

if status = '1' then
do;

  if first.ctrlnm13 then
    do;
      zinc = 0;
      salinc = 0;
      zinc2 = 0;
      zadult = 0;
      hhldkid = 0;
      hhkdu18 = 0;
      skdu18 = 0;
      skidu6 = 0;
      skd617 = 0;
      adltdkd29 = 0;
      adltdkd30 = 0;
      kid013 = 0;
      kid515 = 0;
      kid = 0;

```

```

kidu6 = 0;
kid617 = 0;
elder = 0;
hhage = 0;
hhsex = '';
hhspan = '';
hhrace = '';
hhmove = 0;
hhmar = '';
hhgrad = 0;
hhrel = 0;
spouse = '';
nonrel = 0;
coown = 0;
nrldgr = 0;
unrlkid = 0;
unrlper = 0;
parent = '';
grandpa = '';
grankid = '';
refper = '';
relfam = 0;
relhbage = 0;
reloth = 0;
othrel = 0;
nrel2 = '';
nrel3 = '';
norel3 = 0;
scndry = 0;
respnt = '';
hrmar = '';
movedly = '';

if (-999998 <= vother <= 999998) then
do;
  zinc+vother;
  zinc2+vother;
end;
end;

if hhmem='1' then
do;

/* zinc recode - income of reference person and household
members related to reference person
salinc recode - salary income used to calculate poverty level */

```

```

if (age >= 16 and (rel = 1 or rel = 2 or rel = 20 or (rel >= 22 and
    rel <= 26))) then
do;
    zinc+sal;
    salinc+sal;
end;

/* zinc2 recode - household income */
if (age >= 16) then
    zinc2+sal;

/* zadult recodes - COMP03 - number of adults over 18 */
if (age >= 18) then
    zadult+1;

/* the following recodes all have to do with the children in the household */
if rel = 22 then
do;

/* hhldkid recode -- used in setting the GEN3 recode */
hhldkid+1;

/* hhkdu18 recode -- used in POOR recode when accessing the poverty lookup
table */
if 0 <= age < 18 then
    hhkdu18+1;

/* skdu18 recode - OWNC01 - counts the number of kids under the age of 18 */
if ((mar = '6' and 13 < age < 18) or 0 <= age <= 13) then
do;
    skdu18+1;

/* skidu6 recode - OWNC01 - counts the number of kids under the age of 6 */
if 0 <= age < 6 then
    skidu6+1;

/* skd617 recode - OWNC01 - counts the number of kids between 6 and 17 */
if 6 <= age <= 17 then
    skd617+1;
end;

/* adltd29 recode - COMP05 - counts the number of adult single children
between the ages of 18 and 29 */
if 18<=age<=29 and (mar ne '1' and mar ne '2') then
    adltd29+1;

```

```

/* adltd30 recode - COMP05 - counts the number of adult single children older
than 29 */
  if age > 29 and (mar ne '1' and mar ne '2') then
    adltd30+1;

end;      /* end of rel = 22 check */

/* kid013 recode - NQUA05 - increments if anyone between the ages of 0 and 13
is found in a household */
  if (0 le age le 13) then
    kid013+1;

/* kid515 recode - NQUA05 - increments if anyone between the ages of 5 and 15
is found in a household */
  if (5 le age le 15) then
    kid515+1;

/* kid recode - COMP03 and COMP04 - number of unmarried kids in a hhld */
  if ((13 < age < 18 and '3' <= mar <= '6') or ( 0 <= age <= 13)) then
do;
  kid+1;

/* kidu6 recode - COMP03 - counts the number of kids under the age of 6 */
  if 0 <= age < 6 then
    kidu6+1;

/* kid617 recode - COMP03 - counts the number of kids between 6 and 17 */
  if 6 <= age <= 17 then
    kid617+1;

end; /* end of check for single kids under 18 */

/* elder recode - AGE01 - number of elderly persons in a household */
  if (age ge 65) then
    elder+1;

/* hhage and hhsex recode - COMP01,COMP02,BOX02,MOVE04 - set recode to the age
sex span race and move of the householder */
  if (rel = 1 or rel = 2) then
do;
  hhage = age;
  hhsex = sex;
  hhspan = span;
  hhrace = race;
  hhmove = move;
  hhmar = mar;

```

```

hhgrad = grad;
hhrel = rel;
end;

/* spouse recode - COMP05 - whether a spouse is present */
if rel = 20 then
    spouse = 'X';

/* nonrel recode - COMP05 - number of non-relatives */
/* coown recode - COMP05 - number of coowners */
/* nrldgr recode - COMP05 - non-relative lodger */
/* unrelkid recode - COMP05 - number of unrelated children */
/* unrelper recode - COMP05 - number of unrelated persons */
if rel > 26 then
    do;
        nonrel+1;
        if ten = 'X' then
            coown+1;
        else
            if lodsta = '1' then
                nrldgr+1;
            else
                if ten ne 'X' and lodsta ne '1' then
                    do;
                        if 0 <= age <= 17 then
                            unrlnkid+1;
                        else
                            unrlnper+1;
                    end;
                else
                    do;
                        if 0 <= age <= 17 then
                            unrlnkid+1;
                        else
                            unrlnper+1;
                    end;
                end;
            end;
        end;
    end;

/* COMP05 - recodes used to determine three generations in residence */
if rel = 24 then
    do;
        parent = 'X';
        if par gt 0 then
            grandpa = 'X';
    end;
if rel = 23 then

```

```

        grankid = 'X';
        if rel = 1 or rel = 2 then
            refper = 'X';
/* the following code was moved to the lodger section of code ---
        if rel = 20 and par > 0 then
            do;
                sparnt = 'X';
                sprntln = par;
            end;
        if sparnt = 'X' and sprntln = person and par gt 0 then
            sgrndpa = 'X'; */

/* relfam and relhhage recodes - COMP05 */
if famnum >= '2' and famtyp = '3' and famrel = '1' then
    do;
        relfam+1;
        relhhage = age;
    end;

/* reloth recode - COMP05 */
if famrel = '4' then
    reloth+1;

/* othrel recode - COMP05*/
if (rel > 22 and rel <= 26) or (rel=22 and age ge 18) then
    othrel+1;

/* nrel2 - COMP05 - 2 person hhld-not related */
if curper = 2 and rel = 2 then
    nrel2 = 'X';

/* nrel3 - COMP05 - 3 to 8 person hhld-not related, this is the
   first part, the actual recode is set when the last control
   number for the group of persons is found */
if 3 <= curper <= 8 and (rel = 2 or rel = 32 or rel = 34
    or rel = 36 or rel = 38) then
    norel3+1;

/* scndry - COMP05 - one or more secondary families */
if famnum >= '2' and famtyp = '4' then
    scndry+1;

/* respdnt recode -- recode for household respondent who moved in the last
   year -- and also the recode hrmar -- household respondents marital
   status */
if dline1 = pline and '1' <= mvg <= '4' and (datey = move or

```

```

(datey - move = 1 and movm >= datem)) then
do;
  respdnt = 'Y';
  hrmar = mar;
end;

/* movedly recode -- recode for determining if houshldr moved in the last
year */
if (rel = 1 or rel = 2) and (datey = move or (datey - move = 1 and
movm >= datem)) then
  movedly = 'Y';

end; /* end of hhmem check */

end; /* end of status check */

if last.ctrlnm13 then
  do;
    if norel3 = curper then nrel3 = 'X';
    output rcd.perrcd;
    outcnt+1;
  end;

if send then
do;
  put ctrlno '= number of first control numbers processed';
  put ctllst '= number of last control numbers processed';
  put incnt '= number of obs input';
  put outcnt '= number of obs output';
end;

/* end of recode program */
run;

```

```

/* ****
*      S E C T I O N   T W O
*
*      now do the processing not involving the persons data set or
*      any of the other
*      rostered data sets!
*      */
/* ****

```

```

/* ** this section creates a data set with the recodes included. ** */
data rcd.hhrccd (keep = ctrlnm13 phone kitchen zcrowfr zcrowdr zcrowbr
                  rooms confeetr zaq1 zaq2 zadeq maintr zsmhcenr
                  zsmhcmer zsmhcor zsmhcvr zsmhcpr zvir amtxpr
                  amtxvr tprice amtomor amtimor amtwmor amtgmor
                  amttmor amtfmor lrntmor newc saving sjuice
                  sgas soil scoal swood skero ssun sother pubsub);

length ctrlnm13 $13. phone $1 kitchen $1 zcrowfr 3. zcrowdr 3. zcrowbr 3.
      rooms 3. confeetr 3. zadeq $1 zsmhcenr 4. zsmhcmer 4.
      zsmhcmer 4. zsmhcvr 4. zsmhcpr 4. zvir 4. amtxpr 4. amtxvr 4.
      tprice 4. amtomor 3. amtimor 3. amtwmor 3. amtgmor 3. amttmor 3.
      amtfmor 3. lrntmor 3. newc $1 zaq1 3. zaq2 3. maintr 4.
      pubsub $1 saving $1;

merge edt.houshld end=send
      bridg.houshldr(keep=ctrlnm13 phonenum)
      edt.mortg
      rcd.perrcd;
by ctrlnm13;

retain incnt 0;
retain ctrlno 0;
retain ctllst 0;
retain outcnt 0;
retain newcm 0;

incnt+1;
/* if(mod(incnt,500) = 0) then put 'processing obs number ' incnt; */
if first.ctrlnm13 then ctrlno+1;
if last.ctrlnm13 then ctllst+1;
if first.ctrlnm13 then
do;
  phone = 'B';
  kitchen = 'B';
  rooms = .B;
  confeetr = .B;
  zcrowfr = .B;
  zcrowdr = .B;
  zcrowbr = .B;
  tprice = .B;
  zaq1 = .B;
  zaq2 = .B;
  zadeq = 'B';
  maintr = .B;
  zsmhcenr = .B;

```

```

zsmhcbr = .B;
zsmhcbr = .B;
zsmhcbr = .B;
zsmhcbr = .B;
zvir = .B;
amtomor = .B;
amtimor = .B;
amtwmor = .B;
amttmor = .B;
amtgmor = .B;
amtfmor = .B;
lrntmor = .B;
amtxpr = .B;
amtxvr = .B;
newc = 'B';
pubsub = 'B';
saving = 'B';
end;

if status > '0' and status < '4' then
do;

/* phone recode - */
if (status = '1') then do;
  if (phonenum ne '' or telhh = 1 or telav = 1) then phone = '1';
  else if (telhh = 2 and telav = 2) then phone = '2';
  else phone = 'D';
end;

kitchen = '';
/* kitchen recode - EQUP01 -complete kitchen facilities */

if(((nunit2 = '1' or nunit2 = '2' or nunit2 = '4')
  and (sink = '1')
  and (refr = '1')
  and (cook = '1' or burner = '1' or oven = '1'))
  or
  ((nunit2 = '3' or nunit2 = '5')
  and (kexclu = '1')
  and (sink = '1')
  and (refr = '1')
  and (cook = '1' or burner = '1' or oven = '1'))) then
do;
  kitchen = '1';
end;
else

```

```

do;
if ((sink = '2')
    or (refr = '2')
    or (cook = '2' and burner = '2' and oven = '2')
    or ((nunit2 = '3' or nunit2 = '5') and (kexclu = '2'))) then
do;
    kitchen = '2';
end;
end;

/* rooms recode - ROOM01 - number of rooms */

rooms = 0;
rooms = sum(bedrms,kitch,living,dining,famrm,recrm,dens,othfn);

/* temporary fix for rooms until the inventory module can be fixed */
/* if status eq 1 and rooms eq 0 then rooms = 1; */

/* confeet recode - condominium or homeowners assoc. or mobile home monthly
   fees - COOP02 and COST04 */

if confeeq = . then confeet = .;
if confeeq = 0 then confeet = 0;
if confeeq >= 1 then
do;
    confeetr = (confeeq * camf) / 12;
end;

/* zcrowf recode - SQFT02 - units square feet per person */

if status = '1' and (nunit2 = '1' or nunit2 = '4' or nunit2 = '5') then
do;
if unitsf = .D or unitsf = .R or unitsf = . then
    zcrowf = .D;
else
do;
    zcrowfr = unitsf/curper;
end;
end;

/* zcrowd recode - PPRM01 - persons per rooms */

if status = '1' then
do;
    zcrowdr = curper/rooms;
end;

```

```

/* zcrowb recode - PPRM02 - persons per bedrooms */

if status = '1' then
do;
  if bedrms = 0 then
    zcrowbr = 0;
  else
    zcrowbr = curper/bedrms;
end;

end; /* end of status check */

/* tprice recode - PUPR01 - purchase price */
if 1 <= lprice <= 999998 then
  tprice = lprice;
else
  if 1 <= cprice <= 999998 then
    tprice = cprice;
  else
    if lprice = .d or lprice = .r or lprice = . then
      tprice = .d;
    else
      if cprice = .d or cprice = .r or cprice = . then
        tprice = .d;

/* zadeq recode - PROB01 and BOX02B - adequacy of housing */
if status = '1' then
do;
  zadeq = '0';
  zaq1 = 0;
  zaq2 = 0;

/* calculate zaq1 */
if leak = '1' then
  zaq1+1;
if ileak = '1' then
  zaq1+1;
if holes = '1' then
  zaq1+1;
if cracks = '1' then
  zaq1+1;
if bigp = '1' then
  zaq1+1;
if rats = 'X' then
  zaq1+1;

```

```

/* calculate zaq2 */
if ltsok eq '4' or ltsok = '5' then
    zaq2+1;
if badstep = '2' then
    zaq2+1;
if railok = '1' or railok = '3' then
    zaq2+1;
if climb ge 3 and elev ne '2' then
    zaq2+1;

if (baths < 2 and (hotpip = '2'
    or tub = '2'
    or toilet = '2'
    or sharpf = '1'))
    or
    (freeze = '1'
    and (numcold >= '3' and numcold <= '8'))
    or
    (buye = '1')
    or
    (nowire = '2'
    and plugs = '2'
    and (numblow >= '3' and numblow <= '8'))
    or
    (zaq1 ge 5)
    or
    (zaq2 = 4) then
do;
    zadeq = '3';
end;

if zadeq ne '3' then
do;
    if (numltl >= '3' and numltl <= '8') or
        (hequip = 7) or
        (zaq1 = 3 or zaq1 = 4) or
        (zaq2 = 3) or
        (kitchen = '2') then
            zadeq = '2';
    end;

    if zadeq ne '3' and zadeq ne '2' then
        zadeq = '1';
end; /* end of status check for zadeq universe */

```

```
/* recode for maint - MTNC01 - routine maintenance in last year */
```

```
if cstmnt >= 0 then
do;
  maintr = cstmnt / 12;
end;
else
  maintr = cstmnt;
```

```
/* zsmhcen - monthly housing costs recode (without maintenance)
   - COST01 and BOX07 */
```

```
if amtxre < 0 then amtxre = 0;
if mhotfe < 0 then mhotfe = 0;
if cstmnt < 0 then cstmnt = 0;

if (status = '1' and (tenure = '1' or tenure = '2')
   and prent ne 0) then
do;
```

```
zsmhcnr = 0;
```

```
if (1 <= amte <= 998) then zsmhcnr+amte;
if (1 <= amtg <= 998) then zsmhcnr+amtg;
if (amto >= 4) then zsmhcnr+(amto / 12);
if (amtf >= 4) then zsmhcnr+(amtf / 12);
if (amtt >= 4) then zsmhcnr+(amtt / 12);
if (amtw >= 4) then zsmhcnr+(amtw / 12);
```

```
if ((amtxq >= 0) and (taxpmt ne '1' and txpmt2 ne '1')) then
do;
  zsmhcnr+((amtxq - amtxre) / 12);
end;
```

```
if (amti >= 1) and (inspmt ne '1' and inpmt2 ne '1') then
do;
  zsmhcnr+(amti / 12);
end;
```

```
if (confeeq >= 1) then
  zsmhcnr+((confeeq*camf)/12);
if ( 1 <= lrent <= 1996) then
  zsmhcnr+((lrent * flrent) / 12);
if (mhotfe >= 1) then
```

```

zsmhcnr+((mhotfe * fmhotf) / 12);
if ((rent >= 2) and (prent < 1 or prent > 9997)) then
  zsmhcnr+((rent*frent)/12);
if (prent >= 1 and frent >= 1) then
  zsmhcnr+((prent*frent) / 12);
if (pmt >= 1) then
  zsmhcnr+pmt;
if (pmt2 >= 1) then
  zsmhcnr+pmt2;
if (pmt3 >= 1) then
  zsmhcnr+pmt3;
if (pmt4 >= 1) then
  zsmhcnr+pmt4;
/* if (hepmt1 >= 1) then
   zsmhcnr+hepmt1; */
/* if (hepmt2 >= 1) then
   zsmhcnr+hepmt2; */
/* if (hepmt3 >= 1) then
   zsmhcnr+hepmt3; */
if (amtm >= 1) then
  zsmhcnr = zsmhcnr-(amtm / 12);
if (amtm2 >= 1) then
  zsmhcnr = zsmhcnr-(amtm2 / 12);

end; /* end of universe check for zsmhc */

/* zsmhcm - monthly housing costs recode (with maintenance) - COST08
   and COST09 */

if (status = '1' and tenure = '1')then
do;

zsmhcmr = 0;

if (1 <= amte <= 998) then zsmhcmr+amte;
if (1 <= amtg <= 998) then zsmhcmr+amtg;
if (amto >= 4) then zsmhcmr+(amto / 12);
if (amtf >= 4) then zsmhcmr+(amtf / 12);
if (amtt >= 4) then zsmhcmr+(amtt / 12);
if (amtw >= 4) then zsmhcmr+(amtw / 12);

if ((amtxq >= 0) and (taxpmt ne '1' and txpmt2 ne '1')) then
do;
  zsmhcmr+((amtxq - amtxre) / 12);
end;

```

```

if (amti >= 1) and (inspmt ne '1' and inpmt2 ne '1') then
do;
  zsmhcmr+(amti / 12);
end;

if (confeeq >= 1) then
  zsmhcmr+((confeeq*camf)/12);
if ( 1 <= lrent <= 1996) then
  zsmhcmr+((lrent * flrent) / 12);
if (mhotfe >= 1) then
  zsmhcmr+((mhotfe * fmhotf) / 12);
if (pmt >= 1) then
  zsmhcmr+pmt;
if (pmt2 >= 1) then
  zsmhcmr+pmt2;
if (pmt3 >= 1) then
  zsmhcmr+pmt3;
if (pmt4 >= 1) then
  zsmhcmr+pmt4;
/* if (hepmt1 >= 1) then
   zsmhcmr+hepmt1; */
/* if (hepmt2 >= 1) then
   zsmhcmr+hepmt2; */
/* if (hepmt3 >= 1) then
   zsmhcmr+hepmt3; */
if (amt m >= 1) then
  zsmhcmr = zsmhcmr-(amt m / 12);
if (amt m2 >= 1) then
  zsmhcmr = zsmhcmr-(amt m2 / 12);
if ((cstmnt / 12) >= 1) then
  zsmhcmr+(cstmnt / 12);

end; /* end of universe check for zsmhcm */

/* zsmhco - monthly housing costs recode - COST08 and COST09 */

if (status = '1' and tenure = '1') then
do;

  zsmhcor = 0;

  if (1 <= amte <= 998) then zsmhcor+amte;
  if (1 <= amt g <= 998) then zsmhcor+amt g;
  if (amt o >= 4) then zsmhcor+(amt o / 12);
  if (amt f >= 4) then zsmhcor+(amt f / 12);
  if (amt t >= 4) then zsmhcor+(amt t / 12);

```

```

if (amtw >= 4) then zsmhcor+(amtw / 12);

if ((amtxq >= 0) and (taxpmt ne '1' and txpmt2 ne '1')) then
do;
  zsmhcor+((amtxq - amtxre) / 12);
end;

if (amti >= 1) and (inspmt ne '1' and inpmt2 ne '1') then
do;
  zsmhcor+(amti / 12);
end;

if (confeeq >= 1) then
  zsmhcor+((confeeq*camf)/12);
if ( 1 <= lrent <= 1996) then
  zsmhcor+((lrent * flrent) / 12);
if (mhotfe >= 1) then
  zsmhcor+((mhotfe * fmhotf) / 12);
if (pmt >= 1) then
  zsmhcor+pmt;
/* if pmt = 0 and hepmt1 >= 1 then
   zsmhcor+hepmt1; */
if (amtm2 >= 1) then
  zsmhcor = zsmhcor-(amtm2 / 12);

end; /* end of universe check for zsmhco */

/* zsmhcv - monthly housing costs recode - occupied and vacant-for rent
units/ no maintenance - COST02 */

if ((status = '1' and (tenure = '1' or tenure = '2')
  and prent ne 0) or (status = '3' and (vacancy = 1 or
  vacancy = 2) and rent ne 1)) then
do;
  zsmhcvr = 0;

  if (1 <= amte <= 998) then zsmhcvr+amte;
  if (1 <= amtg <= 998) then zsmhcvr+amtg;
  if (amto >= 4) then zsmhcvr+(amto / 12);
  if (amtf >= 4) then zsmhcvr+(amtf / 12);
  if (amtt >= 4) then zsmhcvr+(amtt / 12);
  if (amtw >= 4) then zsmhcvr+(amtww / 12);

  if ((amtxq >= 0) and (taxpmt ne '1' and txpmt2 ne '1')) then
    zsmhcvr+((amtxq - amtxre) / 12);

```

```

if (amti >= 1) and (inspmt ne '1' and inpmt2 ne '1') then
  zsmhcvr+(amti / 12);
if (confeeq >= 1) then
  zsmhcvr+((confeeq*camf)/12);
if ( 1 <= lrent <= 1996) then
  zsmhcvr+((lrent * flrent) / 12);
if (mhotfe >= 1) then
  zsmhcvr+((mhotfe * fmhotf) / 12);
if ((tenure = 2) and ((rent >= 1) and (prent < 1 or prent > 9997))) then
  zsmhcvr+((rent*frent)/12);
if (vacancy = 1 or vacancy = 2) and (rent >= 2) then
  zsmhcvr +((rent*frent)/12);
if (prent >= 1 and frent >= 1) then
  zsmhcvr+((prent*frent) / 12);
if (pmt >= 1) then
  zsmhcvr+pmt;
if (pmt2 >= 1) then
  zsmhcvr+pmt2;
if (pmt3 >= 1) then
  zsmhcvr+pmt3;
if (pmt4 >= 1) then
  zsmhcvr+pmt4;
/* if (hepmt1 >= 1) then
   zsmhcvr+hepmt1; */
/* if (hepmt2 >= 1) then
   zsmhcvr+hepmt2; */
/* if (hepmt3 >= 1) then
   zsmhcvr+hepmt3; */
if (amtma >= 1) then
  zsmhcvr = zsmhcvr-(amtma / 12);
if (amtmb >= 1) then
  zsmhcvr = zsmhcvr-(amtmb / 12);

end; /* end of universe check for zsmhcv */

/* recode for zsmhcp - monthly housing costs as a percent of income '
 - COST03 */

if status = '1' and (tenure = '1' or tenure = '2') and prent ne 0 then
do;
  zsmhcp = 0;
  if zincn ge 1 then
    do;
      zsmhcp = ((zsmhcnr * 12) / zincn) * 100;
    end;
  else if zincn < 1 and zinc ge 1 then

```

```

do;
zsmhcpr = ((zsmhcpr * 12) / zinc) * 100;
end;
else
do;
zsmhcpr = .X;
end;

end; /* end of universe check for zsmhcp */
/* recode for zvi - RATIO1 - value as a percent of income */

if (status = '1' and tenure = '1') or ((status = '2' or status = '3') or
(vacancy = 3 or vacancy = 5)) then
do;

zvir = 0;
if zincn ge 1 then
do;
zvir = value / zincn;
end;
else if zincn < 1 and zinc ge 1 then
do;
zvir = value / zinc;
end;
else
do;
zvir = .;
end;

end; /* end of universe check for zvi */

/* recode for amtomo - monthly cost for amto */
if status = '1' and (buyo < '1' or buyo > '3') then
do;
amtomor = amto / 12;
end;

/* recode for amtimo - monthly cost for amti */
if buyi = '1' then
do;
amtimor = amti / 12;
end;

/* recodes for monthly costs for water (AMTWMO), trash (AMTTMO), bottled
gas (AMTGMO), and other fuel (AMTFMO) - COST06 */

```

```

if amtw ge 4 then
do;
amtwmor = amtw / 12;
end;
if amtt ge 4 and billwt ne 'X' then
do;
amttmor = amtt / 12;
end;
if amtg ge 1 and gaspip = '2' and billtg ne 'X' and billwg ne 'X' then
do;
amtgmor = amtg;
end;
if amtf ge 4 and billtf ne 'X' and billwf ne 'X' then
do;
amtfmor = amtf / 12;
end;

/* recode for monthly cost for land rent (LRNTMO) - COST04 */
if (incs = 1 and lanpmt = '2') or (incs = 1 and mg = '2') then
do;
if flrent >= 0 and lrent >= 0 then
do;
lrntmor = (flrent * lrent) / 12;
end;
else
lrntmor = 0;
end; /* universe check for lrntmo */
/* amtxp recode - monthly real estate taxes - TAXS01 */

if((status = '1' and tenure = '1') or ((status = '2' or status = '3') and
(vacancy = 3 or vacancy = 5))) then
do;
amttxpr = .;
if ((status = '1' and tenure = '1') and txre ne '1' and amtxq = 0) then
amttxpr = 0;
else
if((status = '1' and tenure = '1') and txre = '1' and
((amtqx - amttxre) <= 0)) then
amttxpr = 0;
else
if((status = '2' or status = '3') and (vacancy = 3 or vacancy = 5)
and amtxq <= 0) then
amttxpr = 0;
else
if((status = '1' and tenure = '1') and txre ne '1' and amtxq >= 1)
then

```

```

        amtxpr = amtxq / 12;
else
  if((status = '1' and tenure = '1') and txre = '1' and
    ((amtxq - amtxre) >= 1)) then
    amtxpr = (amtxq - amtxre) / 12;
  else
    if((status = '2' or status = '3') and (vacancy = 3 or
      vacancy = 5) and amtxq >= 1) then
      amtxpr = amtxq / 12;

end; /* end of universe check for amtx */

/* recode for amtxv - real estate taxes as a percent of value - TAXS02 */
if (status = '1' and tenure = '1') or ((status = '3' or status = '2') and
  (vacancy = 3 or vacancy = 5)) then
do;
  amtxvr = (amtxpr * 12) / (value / 1000);

end;

/* recode for (Total Units) Publication -- NEWC - New construction last
4 years - BOX02 */

newc = ' ';
if '1' <= status <= '3' then
do;
  if (datey - 3) <= built <= datey then
    newc = '1';
  else
    if built = (datey - 4) then
      do;
        if nunit2 = '4' or nunit2 = '5' then
          do;
            if ranuni (datey) <= .25 then
              newc = '1';
            end;
          else
            if mbuilt ge datem then
              newc = '1';
            end;
        end;
      end;
    end;
  end;
end; /* end of universe check for newc */

```

```

/* Recode for PUBSUB -- RENT02 -- Rent reductions */

if (status = '1' and (tenure = '2' or tenure = '3')) or
((status = '2' or status = '3') and (vacancy = 1 or
vacancy = 2 or vacancy = 4)) then
do;
  pubsub = '';
  if proj = '1' then
    pubsub = '6';
  else
    if (subrnd = '1' and proj ne '1') or vcher = '1' or
      apply = '2' then
      pubsub = '7';
    else
      if renew = '1' and (reph = '1' or repa = '2') then
        pubsub = '8';
      else
        if rcntrl = '1' then
          pubsub = '1';
        else
          if subrnd = '2' and (rcntrl = 'D' or rcntrl = 'R' or
            rcntrl = ' ') then
              pubsub = '5';
            else
              if rcntrl = '2' and rntadj = '1' then
                pubsub = '2';
              else
                if rcntrl = '2' and rntadj = '2' then
                  pubsub = '3';
                else
                  if rcntrl = '2' and ((rntadj = 'D' or rntadj = 'R' or
                    rntadj = ' ') or ((status = '2' or status = '3') and
                      (vacancy = 1 or vacancy = 2 or vacancy = 4) and (rntadj = 'B')))) then
                      pubsub = '4';
                    else
                      pubsub = '9';
end; /* end of universe check for pubsub */

```

```

/* Recode for saving -- SAVI01 -- Savings and Investments */

```

```

if -9999999 <= zinc <= 25000 then
do;
  if qsavng = '2' and qbinv = '2' and qoinv = '2' then
    saving = '1';
  else
    if inv20k = '2' then

```

```

    saving = '2';
else
  if inv20k = '1' then
    saving = '3';
  else
    saving = '4';
end;

if send then
do;
  put ctrlno ' = number of first control numbers processed';
  put ctllst ' = number of last control numbers processed';
  put incnt ' = number of view obs input';
  put outcnt ' = number of obs output';
end;
run;

/* **** */
/*      SECTION THREE          */
/* **** */
/* Calculate POOR by using a poverty look up table      */
/* **** */
libname library "$RECD";

data rcd.povrcd (keep=ctrlnm13 poorr pvlkup povlvl);
length poorr 4. povlvl 4.;

set rcd.perrcd;
  by ctrlnm13;

attrib pvlkup format=$5.
  agehh pers kids format=$1.
  imonth format=$2.;

if status = '1' then
do;

/* calculate the lookup variable to retrieve the poverty threshold */
if (curper = 1 or curper = 2) then
  do;
    pers=put(curper,1.);
    if 0 <= hhage < 65 then agehh='1';
    else if hhage ge 65 then agehh='2';
  end;

```

```

else if curper > 2 then
do;
  agehh='3';
  if curper >= 9 then
    pers='9';
  else if curper < 9 then
    pers=put(curper,1.);
end;

if hhkdu18 ge 8 then kids='8';
else if 0 <= hhkdu18 < 8 then kids=put(hhkdu18,1.);

if datem <= 8 then imonth='08';
else if datem => 11 then imonth='11';
else if datem = 9 then imonth='09';
else if datem = 10 then imonth='10';

pvlkup = imonth||pers||agehh||kids;

/* retrieve poverty threshold level from the previously created poverty
* formats */

povlvl=input(put(pvlkup,$povfmt.),6.);

/* calculate the income as percent of poverty level */

if povlvl ne 0 then
do;
  if zinc2 ne 0 then
    do;
      poorr=(zinc2*1000)/(povlvl);
    end;
    else poorr=1;
  end;
  else
    put ' error in poverty lookup for ctrlnm13 = ' ctrlnm13;
end;

/* check end ranges for the calculated poor variable */

if poorr lt 1 then
  poorr=1;

end; /* status check */
run;

/* **** */

```

```

/*      S E C T I O N      F O U R      */
/*
/* This section calculates the inc3flg for the INCM03 stub      */
/* **** */
data rcd.inc3rcd (keep=ctrlnm13 salinc inc3flg zinc status curper
per20wg)
rcd.inc3rcdp (keep=ctrlnm13 salinc per20wg sal age status curper
rel);

merge rcd.perrcd (in=inc3)
edt.houshld (keep = ctrlnm13 status curper)
edt.person (in=pers);
by ctrlnm13;

length inc3flg $1;
retain per20wg 0;
retain cntr 0;

if status = '1' then
do;
  if first.ctrlnm13 then
    do;
      per20wg = salinc * .2;
      cntr = 0;
      inc3flg = '';
      end;
    if (age ge 16 and ((1<=rel<=2) or rel = 20 or (22 <= rel <=26))) then
      do;
        if (sal gt 0 and sal gt per20wg) then
          cntr+1;
        end;
      end; /* end of status check */

    if last.ctrlnm13 then
      do;
        if cntr > 1 then
          inc3flg = 'X';
        else
          inc3flg = '';
        output rcd.inc3rcdp;
        output rcd.inc3rcd;
      end;
    else
      output rcd.inc3rcdp;
  end;
end;

```

```

run;

/* *****
*      S E C T I O N   F O U R
*
*      The following code recodes the mortgage variables
*      OTPIN - outstanding principle and interest
*      CLPEVA - current loan as percent of value
*      CRINTR - current interest rate
*
***** */
*/



data rcd.mtgrcd (keep = ctrlnm13 otpinr clpevar crintrr);

length otpinr 4. clpevar 4. crintrr 4.;

merge edt.houshld
      edt.mortg;
by ctrlnm13;

otpinr = .B;
clpevar = .B;
crintrr = .B;

if status = '1' then
  do;

/* recode for otpin */

otpin1 = 0;
otpin2 = 0;
otpin3 = 0;
otpin4 = 0;

if mcnt ge 1 then
do;
  otpinr = 0;
  p1 = ammort;
  j1 = 1 + (.01 * ((intw + (intf * .125)) + .125) / 12);
  if matbuy = '1' then
    do;
      k1 = ((12*(datey-whnget))-6)+datem;
      if k1 le 0 then
        k1 = 1;

```

```

if newmor = '1' or newmor = '3' then
do;
  if 15 <= term <= 41 then
    n1 = term*12;
    if (1<=term<=14 or canvar = 'X') and 1<=amrtz<=41 then
      n1 = amrtz*12;
    if (1 <= term <= 14 or canvar = 'X') and (amrtz < 1 or amrtz > 41)
    then
      n1 = .D;
  end;
  if newmor = '2' then
do;
  n1 = term*12;
end;
end;

if matbuy = '2' then
do;
  if 1900 <= yrmor <= datey then
do;
  k1 = ((12*(datey-ymor))-6)+datem;
  if k1 le 0 then
    k1 = 1;
  end;
  if yrmor < 1900 or yrmor > datey then
    k1 = .D;
  if 15 <= term <= 41 then
    n1 = term*12;
    if (1 <= term <= 14 or canvar = 'X') and 1 <= amrtz <= 41 then
      n1 = amrtz*12;
    if (1 <= term <= 14 or canvar = 'X') and (amrtz < 1 or amrtz > 41) then
      n1 = .D;
  end;
  if k1 eq .D or n1 = .D then
    otpin1 = .D;
  if k1 ne .D and n1 ne .D then
    do;
      /* if k1 ge n1 then k1 = n1 - 1; */
      otpin1 = (1-(j1**(k1-n1)))/(1-(j1**(-n1)))*p1;
    end;
end;

if mcnt ge 2 then
do;
  p2 = ammrt2;
  j2 = 1 + (.01 * ((intw2 + (intf2 * .125)) + .125) / 12);

```

```

if matby2 = '1' then
do;
  k2=((12*(datey-whnget))-6)+datem;
  if k2 <= 0 then
    k2 = 1;
  if newmr2 = '1' or newmr2 = '3' then
    do;
      if 15 <= term2 <= 41 then
        n2 = term2*12;
      if (1 <= term2 <= 14 or canvr2 = 'X') and 1 <= amrtz2 <= 41 then
        n2 = amrtz2*12;
      if (1 <= term2 <= 14 or canvr2 = 'X') and (amrtz2 < 1 or
          amrtz2 > 41) then
        n2 = .D;
      end;
    if newmr2 = '2' then
      do;
        n2 = term2*12;
      end;
    end;
  if matby2 = '2' then
    do;
      if 1900 <= yrmor2 <= datey then
        do;
          k2 = ((12*(datey-ymor2))-6)+datem;
          if k2 le 0 then
            k2 = 1;
        end;
      if yrmor2 < 1900 or yrmor2 > datey then
        k2 = .d;
      if 15 <= term2 <= 41 then
        n2 = term2*12;
      if (1 <= term2 <= 14 or canvr2 = 'X') and 1 <= amrtz2 <= 41 then
        n2 = amrtz2*12;
      if (1 <= term2 <= 14 or canvr2 = 'X') and (amrtz2 < 1 or
          amrtz2 > 41) then
        n2 = .D;
      end;
    if k2 = .D or n2 = .D then
      otpin2 = .D;
    if k2 ne .D and n2 ne .D then
      do;
        /* if k2 ge n2 then k2 = n2 - 1; */
        otpin2 = (1-(j2**(k2-n2)))/(1-(j2**(-n2)))*p2;
      end;
    end;
end;

```

```
if mcnt ge 3 then
do;
  if 1 <= ammrt3 <= 999998 then
    otpin3 = .75*ammrt3;
  if ammrt3 < 1 or ammrt3 > 999998 then
    otpin3 = 0;
end;
```

```
if mcnt ge 4 then
do;
  if 1 <= ammrt4 <= 999998 then
    otpin4 = .75*ammrt4;
  if ammrt4 < 1 or ammrt4 > 999998 then
    otpin4 = 0;
end;
```

```
if 1 <= mcnt <= 7 then
do;
  if k1 = .D or n1 = .D or k2 = .D or n2 = .D then
    otpinr = .D;
  else
    do;
      otpinr = otpin1 + otpin2 + otpin3 + otpin4;
      if otpinr ne .D and otpinr lt 0 then otpinr = 0;
    end;
  end;
```

```
/* recode for clpeva */
if 1 <= mcnt <= 7 then
do;
  clpevar = .;
  if otpinr ne .D then
    do;
      clpevar = otpinr / value * 100;
    end;
  if otpinr = .D then
    clpevar = .D;
end;
```

```
/* recode for crintr */
```

```
if 1 <= mcnt <= 7 then
do;
```

```

crinrr = .;
if mcnt = 1 then
do;
  crinrr = intw + (intf/8);
end;
if mcnt => 2 then
do;
  if otpin1 ne .D and otpin2 ne .D then
    do;
      crinrr = (((intw+(intf/8))*otpin1)+((intw2+(intf2/8))*otpin2))/
        (otpin1+otpin2);
    end;
  if otpin1 = .D or otpin2 = .D then
    crinrr = .D;
end;
end;

end; /* end of status check */
run;

```

```

/* ****
*          S E C T I O N   F I V E
*
* Second part of mortgage recodes including
*   RYMORT -- remaining years mortgaged
*   PRIN01 -- monthly payment for principle and interest
*   PAYPM1 -- payment plan for first mortgage
*   PAYPM2 -- payment plan for second mortgage
*   CREDLP -- type of home equity loans
***** */

```

```
data rcd.mtg2rcd (keep = ctrlnm13 rymort prin01r paypm1 paypm2 credlp);
```

```
length rymort 4. paypm1 $1 paypm2 $1 prin01r 4. credlp $1;
```

```
merge edt.houshld
  edt.mortg;
by ctrlnm13;
```

```
prin01r = .;
```

```
if status = '1' then
do;
```

```

/* recode for rymort - MORT10 - remaining years mortgaged */
if 1 <= mcnt <= 7 and canvar ne 'X' and canvr2 ne 'X' then
do;
if mcnt = 1 then
do;
if matbuy = '1' then
    rymort = term - (datey-whnget);
if matbuy = '2' then
    rymort = term - (datey-yrmor);
end;

if 2 <= mcnt <= 7 then
do;
if matbuy = '1' then
    rymort1 = (datey - whnget);
if matbuy = '2' then
    rymort1 = (datey-yrmor);
if matby2 = '1' then
    rymort2 = term2 - (datey - whnget);
if matby2 = '2' then
    rymort2 = term2 - (datey - yrmor2);
if rymort1 >= rymort2 then
    rymort = rymort1;
if rymort1 < rymort2 then
    rymort = rymort2;
if rymort < 0 then
    rymort = 0;
end;
end;

```

/* recode for prin01 -- monthly payment for principal and interest */

```

if tenure = '1' and 1 <= mcnt <= 7 then
do;
/* prin01 = 0; */
prin01r = 0;
if mcnt = 1 then prin01r+pmt;
if mcnt = 2 then prin01r+pmt+pmt2;
if mcnt = 3 then prin01r+pmt+pmt2+pmt3;
if 4 <= mcnt <= 7 then prin01r+pmt+pmt2+pmt3+pmt4;
if (taxpmt = '1' or txpmt2 = '1') and 1 <= amtxq <= 99998 then
    prin01r = prin01r - (amtxq / 12);
if (inspmt = '1' or inpmt2 = '1') and 1 <= amti <= 9998 then
    prin01r = prin01r - (amti / 12);
if othpmt = '1' and 1 <= amtm <= 999998 then
    prin01r = prin01r - (amtm / 12);

```

```

if otpmt2 = '1' and 1 <= amtm2 <= 999998 then
    prin01r = prin01r - (amtm2 / 12);
if prin01r < 0 then
    prin01r = 0;
end;

/* recode for paypm1 - 15 - payment plan of primary mortgage */

if mcnt < 1 or mcnt > 7 then
    paypm1 = 'B';
if 1 <= mcnt <= 7 then
do;
    paypm1 = 'D';
    count = 0;

if arm = 'X' then count+1;
if gpm = 'X' then count+1;
if gpmw = 'X' then count+1;
if bloon = 'X' then count+1;
if varm = 'X' then count+1;

if (count = 2 and (gpm ne 'X' or gpmw ne 'X')) or count > 2 then
    paypm1 = '7';
else
if matbuy = '2' or newmor = '1' or newmor = '3' then
do;
    if ((1 <= term <= 14 and term >= amrtz) or 15 <= term <= 41)
        and
        ((vary = '1') or (fixed = 'X' and arm ne 'X' and gpm ne 'X'
        and gpmw ne 'X' and bloon ne 'X' and varm ne 'X')) then
            paypm1 = '1';
    else
        if ((1 <= term <= 14 and term >= amrtz) or 15 <= term <= 41)
            and
            (arm = 'X' and gpm ne 'X' and gpmw ne 'X' and bloon ne 'X'
            and varm ne 'X') then
                paypm1 = '2';
    else
        if canvar = 'X' and (vary = '1' or (fixed = 'X' and arm ne 'X'
        and gpm ne 'X' and gpmw ne 'X' and bloon ne 'X' and varm ne
        'X')) then
            paypm1 = '3';
    else
        if ((1 <= term <= 14 and term < amrtz) or 15 <= term <= 41)
            and
            (arm ne 'X' and (gpm = 'X' or gpmw = 'X') and bloon ne 'X' and
            varm ne 'X') then
                paypm1 = '4';
        else
            paypm1 = '5';
    end;
end;

```

```

varm ne 'X') then
    paypm1 = '4';
else
if (1 <= term <= 14 and term < amrtz and (vary = '1' or ((fixed =
    'X' or bloon = 'X') and arm ne 'X' and gpm ne 'X' and gpmw
    ne 'X' and varm ne 'X')))
or
(1 <= term <= 41 and bloon = 'X' and arm ne 'X' and gpm ne 'X'
and gpmw ne 'X' and varm ne 'X') then
    paypm1 = '5';
else
if (1 <= term <= 41 and term < amrtz)
    and
    (arm ne 'X' and gpm ne 'X' and gpmw ne 'X' and bloon ne 'X'
    and varm = 'X') then
        paypm1 = '6';
else
if (canvar = 'X' and (arm = 'X' or gpm = 'X' or gpmw = 'X' or
    bloon = 'X' or varm = 'X'))
or
((1 <= term <= 14 and term < amrtz) and (arm = 'X' or gpm = 'X'
or gpmw = 'X' or varm = 'X')) then
    paypm1 = '7';
else
if (vary = 'D' or vary = 'R' or vary = ' ')
or
(vary = '2' and fixed ne 'X' and arm ne 'X' and gpm ne 'X'
and gpmw ne 'X' and bloon ne 'X' and varm ne 'X')
or
(vary = '2' and fixed ne 'X' and arm ne 'X' and gpm ne 'X'
and gpmw ne 'X' and bloon ne 'X' and varm ne 'X') then
    paypm1 = 'D';
end; /* matbuy and newmor check */
else
if newmor = '2' then
do;
    if vary = '1' or (fixed = 'X' and arm ne 'X' and gpm ne 'X' and
        gpmw ne 'X' and bloon ne 'X' and varm ne 'X') then
            paypm1 = '1';
    else
        if bloon = 'X' and arm ne 'X' and gpm ne 'X' and gpmw ne 'X'
            and varm ne 'X' then
                paypm1 = '5';
    else
        if (vary = 'D' or vary = 'R' or vary = ' ')
        or

```

```

(vary = '2' and fixed ne 'X' and arm ne 'X' and gpm ne 'X' and
gpmw ne 'X' and varm ne 'X')
or
(vary = '2' and fixed ne 'X' and arm ne 'X' and gpm = 'X' and
gpmw = 'X' and bloon ne 'X' and varm ne 'X') then
    paypm1 = 'D';
end; /* newmor = 2 check */
end; /* mcnt = 1 - 7 check for paypm1 recode */

/* paypm2 recode - mort16 - payment plan of secondary mortgage */
if mcnt < 2 or mcnt > 7 then
    paypm2 = 'B';
if 2 <= mcnt <= 7 then
do;
    paypm2 = 'D';
    count2 = 0;
    if arm2 = 'X' then count2+1;
    if gpm2 = 'X' then count2+1;
    if gpmw2 = 'X' then count2+1;
    if bloon2 = 'X' then count2+1;
    if varm2 = 'X' then count2+1;
    if (count2 = 2 and (gpm2 ne 'X' or gpmw2 ne 'X')) or count2 > 2 then
        paypm2 = '7';
    else
        if matby2 = '2' or newmr2 = '1' or newmr2 = '3' then
            do;
                if ((1 <= term2 <= 14 and term2 >= amrtz2) or 15 <= term2 <= 41)
                    and
                    (vary2 = '1' or (fixed2 = 'X' and arm2 ne 'X' and gpm2 ne 'X' and
gpmw2 ne 'X' and bloon2 ne 'X' and varm2 ne 'X')) then
                        paypm2 = '1';
                else
                    if ((1 <= term2 <= 14 and term2 >= amrtz2) or 15 <= term2 <= 41)
                        and
                        (arm2 = 'X' and gpm2 ne 'X' and gpmw2 ne 'X' and bloon2 ne 'X'
and varm2 ne 'X') then
                            paypm2 = '2';
                    else
                        if canvr2 = 'X'
                            and
                            (vary2 = '1' or (fixed2 = 'X' and arm2 ne 'X' and gpm2 ne 'X'
and gpmw2 ne 'X' and bloon2 ne 'X' and varm2 ne 'X')) then
                                paypm2 = '3';
                else
                    if ((1 <= term2 <= 14 and term2 >= amrtz2) or 15 <= term2 <= 41)
                        and

```

```

(arm2 ne 'X' and (gpm2 = 'X' or gpmw2 = 'X') ) and bloon2 ne 'X'
and varm2 ne 'X') then
    paypm2 = '4';
else
if (1 <= term2 <= 14 and term2 < amrtz2 and (vary2 = '1' or
((fixed2 = 'X' or bloon2 = 'X') and arm2 ne 'X' and gpm2 ne 'X'
and gpmw2 ne 'X' and varm2 ne 'X')))
or
(1 <= term2 <= 41 and bloon2 = 'X' and arm2 ne 'X' and gpm2
ne 'X' and gpmw2 ne 'X' and varm2 ne 'X') then
    paypm2 = '5';
else
if (1 <= term2 <= 41 and term2 < amrtz2) and (arm2 ne 'X' and
gpm2 ne 'X' and gpmw2 ne 'X' and bloon2 ne 'X' and varm2 = 'X')
then
    paypm2 = '6';
else
if (canvr2 = 'X' and (arm2 = 'X' or gpm2 = 'X' or gpmw2 = 'X' or
bloon2 = 'X' or varm2 = 'X'))
or
((1 <= term2 <= 14 and term2 < amrtz2) and (arm2 = 'X' or
gpm2 = 'X' or gpmw2 = 'X' or varm2 = 'X')) then
    paypm2 = '7';
else
if (vary2 = 'D' or vary2 = 'R' or vary2 = ' ')
or
(vary2 = '2' and fixed2 ne 'X' and arm2 ne 'X' and gpm2 ne 'X'
and gpmw2 ne 'X' and bloon2 ne 'X' and varm2 ne 'X')
or
(vary2 = '2' and fixed2 ne 'X' and arm2 ne 'X' and gpm2 = 'X'
and gpmw2 = 'X' and bloon2 ne 'X' and varm2 ne 'X') then
    paypm2 = 'D';
end; /* end of matby2 and newmr2 check */
else
if newmr2 = '2' then
do;
    if vary2 = '1'
    or
    (fixed2 = 'X' and arm2 ne 'X' and gpm2 ne 'X' and gpmw2 ne 'X'
and bloon2 ne 'X' and varm2 ne 'X') then
        paypm2 = '1';
    else
    if bloon2 = 'X' and arm2 ne 'X' and gpm2 ne 'X' and gpmw2 ne 'X'
and varm2 ne 'X' then
        paypm2 = '5';
    else

```

```

if (vary2 = 'D' or vary2 = 'R' or vary2 = ' ')
  or
  (vary2 = '2' and fixed2 ne 'X' and arm2 ne 'X' and gpm2 ne 'X'
   and gpmw2 ne 'X' and varm2 ne 'X')
  or
  (vary2 = '2' and fixed2 ne 'X' and arm2 ne 'X' and gpm2 = 'X'
   and gpmw2 = 'X' and bloon2 ne 'X' and varm2 ne 'X') then
    paypm2 = 'D';
end; /* end of newmr2 = 2 check */

end; /* end of mcnt = 2-7 check for paypm2 */

/* CREDLP recode -- type of home equity loan */
if hel = '1' then
do;
  if (hetyp1 = '2' and hetyp2 ne '1' and hetyp3 ne '1') or
    (hetyp1 ne '1' and hetyp2 = '2' and hetyp3 ne '1') or
    (hetyp1 ne '1' and hetyp2 ne '1' and hetyp3 = '2') then
      credlp = '1';
  else
    if (hetyp1 = '1' and hetyp2 ne '2' and hetyp3 ne '2') or
      (hetyp1 ne '2' and hetyp2 = '1' and hetyp3 ne '2') or
      (hetyp1 ne '2' and hetyp2 ne '2' and hetyp3 = '1') then
        credlp = '2';
  else
    if (hetyp1 = '1' or hetyp2 = '1' or hetyp3 = '1') and
      (hetyp1 = '2' or hetyp2 = '2' or hetyp3 = '2') then
        credlp = '3';
  else
    credlp = '4';
end; /* end of hel check */

end; /* end of status check */

run;

```

```

/*
*          S E C T I O N   S I X
*
* recent movers recode for table stub MOVE02
*/

```

```

data mrgd1;
merge edt.houshld (keep=ctrlnm13 status datem datey curper almv)
  edt.person (keep=ctrlnm13 move movm rel mvg);

```

```

by ctrlnm13;
run;

proc sort data=mrgd1 out=mrgd2;
  by ctrlnm13 mvg;
run;

data rcd.mv2rcd (keep = ctrlnm13 amvone mv1hh mv2hh amvtwo smv3 mv3hh
                  mv4nr lastyr xheadhh xunithh xcosthh xperhh
                  xinushh xtenhh xrelhh rmvunvh)
  rcd.mv2rec;

length amvone 4. amvtwo 4. smv3 4. mv1hh $1 mv2hh $1 mv3hh $1
      mv4nr $1 xheadhh $1 xunithh $1 xcosthh $1 xperhh 3.
      xinushh $1 xtenhh $1 xrelhh $1 rmvunvh $1 lastyr 3.;

merge mrgd2
  edt.rmov (rename=(movgrp=mvg)) ;
by ctrlnm13 mvg;

retain lastyr 0;
retain lastyrfl '';
retain zeros 0;
retain ones 0;
retain twos 0;
retain xheadhh '';
retain xhead1 '';
retain lstmvg '';
retain xunithh '';
retain xcosthh '';
retain xperhh .;
retain xinushh '';
retain xtenhh '';
retain xrelhh '';
retain rmvunvh '';
retain totmov 0;

if first.ctrlnm13 then
do;
  lastyr = 0;
  amvtwo = 0;
  smv3 = 0;
  amvone = 0;
  lastyrfl = '';
  ones = 0;
  twos = 0;

```

```

zeros = 0;
xheadhh = '';
xhead1 = '';
lstmvg = '';
xunithh = '';
xcosthh = '';
xperhh = 0;
xinushh = '';
xtenhh = '';
xrelhh = '';
rmvunvh = '';
totmov = 0;
end;

if status = '1' then
do;

if (move = datey) or ((move = datey-1) and (datem <= movm)) then
do;
lastyr+1;
lastyrfl = 'X';
if ('1' <= mvg <= '4' and lstmvg ne mvg) then
  totmov+1;
end; /* with in last year check */

if rel = 1 or rel = 2 then
do;
  if xinus ne 'X' and
    ((move = datey) or ((move = datey-1) and (datem <= movm))) then
      rmvunvh = 'X';
end;

if mvg >= '1' and mvg <= '4' then
do;
  if (move = datey) or ((move = datey - 1) and (datem <= movm)) then
    do;
      if lstmvg ne mvg then
        do;
          if (xhead = '' or xhead = 'D' or
              xhead = 'R' or xhead = 'B') then zeros+1;
          if xhead = '1' then ones+1;
          if xhead = '2' then twos+1;
          if mvg = '1' then xhead1 = xhead;
          lstmvg = mvg;
        end;

```

```

if (rel = 1 or rel = 2) then
do;
  xheadhh = xhead;
  xunithh = xunit;
  xcosthh = xcost;
  xperhh = xper;
  xinushh = xinus;
  xrelhh = xrel;
  xtenhh = xten;
end;
end;

end; /* mvg check */

if last.ctrlnm13 and lastyrlf = 'X' then
do;
  if almv = '1' and lastyr = curper then
    do;
      amvone+1;
      if xhead1 = '2' then
        mv1hh = '0';
      if xhead1 = '1' then
        mv1hh = '1';
      if xhead1 = 'D' or xhead1 = 'R' or xhead1 = ''
        or xhead1 = 'B' then
        mv1hh = 'N';
    end;
  else
    if almv = '2' and lastyr = curper then
      do;
        amvtwo+1;
        if twos = totmov then
          mv2hh = '0';
        else
          if ones = 1 and (twos = totmov-1) then
            mv2hh = '1';
          else
            if ones >= 2 then
              mv2hh = '2';
            else
              if zeros ge 1 then
                mv2hh = 'N';
      end;
    else
      if lastyr lt curper then

```

```

do;
  smv3+1;
  if twos = totmov then
    mv3hh = '0';
  else
    if ones ge 1 then
      mv3hh = '1';
    else
      if zeros ge 1 then
        mv3hh = 'N';
    end;
  else
    if lastyr = curper and (almv = ' ' or almv = 'D' or
      almv = 'R' or almv = 'B')then
      mv4nr = 'N';

end; /* end of processing last ctrlnm13 and mvg */

end; /* end of status = 1 check */

output rcd.mv2rec;
if last.ctrlnm13 then output rcd.mv2rcd;

run;
/*
*****
*      S E C T I O N   E I G H T
*
*      Recode section to tally all workers for jtw (journey to work)
*      and to assign housholder variables for jtw
*****
*/

```

```

data jtwsu1;

merge edt.houshld (keep = ctrlnm13 status)
      edt.person (keep = ctrlnm13 person wlineq rel age);
by ctrlnm13;

run;

data rcd.jtwrcd (keep=ctrlnm13 j1drv j1cp j1cp2 j1cp3 j1cp4 j1mass j1taxi
                  j1bike j1walk j1oth j1home j2u15 j21529 j23044 j24559
                  j26089 j2o90 j2home j2noplj3u1 j314 j359 j31019
                  j32029 j33049 j3o50 j3home j3noplj4123a j436a j467a

```

```

j47730a j47308a j48830a j48309a j4910a j4104p j4412p
j4nr hhwlineq hhtran hhvehcl hhpss hhhjob hhtimej
hhdistj hhampm hhwtme jtwtot)

rcd.jtwjobs;
merge jtwsu1
edt.jtw;
by ctrlhm13 person;
length j1drv 3. j1cp 3. j1cp2 3. j1cp3 3. j1cp4 3. j1mass 3. j1taxi 3.
j1bike 3. j1walk 3. j1oth 3. j1home 3. j2u15 3. j21529 3. j23044 3.
j24559 3. j26089 3. j2o90 3. j2home 3. j2noplc 3. j3u1 3. j314 3.
j359 3. j31019 3. j32029 3. j33049 3. j3o50 3. j3home 3. j3noplc 3.
j4123a 3. j436a 3. j467a 3. j47730a 3. j47308a 3. j48830a 3.
j48309a 3. j4910a 3. j4104p 3. j4412p 3. j4nr 3. hhwlineq 3.
hhtran 3. hhvehcl $1 hhpss 3. hhhjob 3. hhtimej 3. hhdistj 3.
hhampm $1 hhwtme $4 jtwtot 3.;

retain j1drv 0;      /* JTW01 */
retain j1cp 0;      /* JTW01 */
retain j1cp2 0;     /* JTW01 */
retain j1cp3 0;     /* JTW01 */
retain j1cp4 0;     /* JTW01 */
retain j1mass 0;    /* JTW01 */
retain j1taxi 0;    /* JTW01 */
retain j1bike 0;    /* JTW01 */
retain j1walk 0;    /* JTW01 */
retain j1oth 0;     /* JTW01 */
retain j1home 0;    /* JTW01 */
retain j2u15 0;     /* JTW02 */
retain j21529 0;    /* JTW02 */
retain j23044 0;    /* JTW02 */
retain j24559 0;    /* JTW02 */
retain j26089 0;    /* JTW02 */
retain j2o90 0;     /* JTW02 */
retain j2home 0;    /* JTW02 */
retain j2noplc 0;   /* JTW02 */
retain j3u1 0;      /* JTW03 */
retain j314 0;      /* JTW03 */
retain j359 0;      /* JTW03 */
retain j31019 0;    /* JTW03 */
retain j32029 0;    /* JTW03 */
retain j33049 0;    /* JTW03 */
retain j3o50 0;     /* JTW03 */
retain j3home 0;    /* JTW03 */
retain j3noplc 0;   /* JTW03 */
retain j4123a 0;    /* JTW04 */
retain j436a 0;     /* JTW04 */

```

```

retain j467a 0;      /* JTW04 */
retain j47730a 0;    /* JTW04 */
retain j47308a 0;    /* JTW04 */
retain j48830a 0;    /* JTW04 */
retain j48309a 0;    /* JTW04 */
retain j4910a 0;     /* JTW04 */
retain j4104p 0;     /* JTW04 */
retain j4412p 0;     /* JTW04 */
retain j4nr 0;       /* JTW04 */
retain hhwlineq .;   /* JTW05 - JTW08 */
retain hhtran .;     /* JTW05 - JTW08 */
retain hhvehcl ' ';  /* JTW05 - JTW08 */
retain hhpss .;      /* JTW05 - JTW08 */
retain hhjob .;      /* JTW05 - JTW08 */
retain hhtimej .;    /* JTW05 - JTW08 */
retain hhdistj .;    /* JTW05 - JTW08 */
retain hhampm ' ';   /* JTW05 - JTW08 */
retain hhwtme ' '/* JTW05 - JTW08 */
retain jtwtot 0;     /* JTW01 - JTW04 */

```

if first.ctrlnm13 then

```

do;
j1drv = 0;
j1cp = 0;
j1cp2 = 0;
j1cp3 = 0;
j1cp4 = 0;
j1mass = 0;
j1taxi = 0;
j1bike = 0;
j1walk = 0;
j1oth = 0;
j1home = 0;
j2u15 = 0;
j21529 = 0;
j23044 = 0;
j24559 = 0;
j26089 = 0;
j2o90 = 0;
j2home = 0;
j2nopl = 0;
j3u1 = 0;
j314 = 0;
j359 = 0;
j31019 = 0;
j32029 = 0;

```

```

j33049 = 0;
j3o50 = 0;
j3home = 0;
j3noplcl = 0;
j4123a = 0;
j436a = 0;
j467a = 0;
j47730a = 0;
j47308a = 0;
j48830a = 0;
j48309a = 0;
j4910a = 0;
j4104p = 0;
j4412p = 0;
j4nr = 0;
hhwlineq = .;
hhtran = .;
hhvehcl = '';
hhpass = .;
hhhjob = .;
hhtimej = .;
hhdistj = .;
hhampm = '';
hhwtime = ' ';
jtwtot = 0;
end;

if status = '1' then
do;
if wlineq = '1' and winus ne '1' and age >= 14 then
do;
jtwtot+1;
if 1 <= tran <=3 then
do;
if vehcl = 1 then j1drv+1;
else
if vehcl = 2 then j1cp+1;
do;
if pass = 2 then j1cp2+1;
else
if pass = 3 then j1cp3+1;
else
if pass >= 4 then j1cp4+1;
end;
end;
else

```

```

if 4 <= tran <= 6 then j1mass+1;
else
  if tran = 7 then j1taxi+1;
else
  if 8 <= tran <= 9 then j1bike+1;
else
  if tran = 11 then j1walk+1;
else
  if tran = 10 then j1oth+1;
else
  if tran = 12 then j1home+1;

if timej = 996 or distj = 996 then
do;
  j2home+1;
  j3home+1;
end;
else
if hjob ne '2' then
do;
  if 0 <= timej <= 14 then j2u15+1;
  else
    if 15 <= timej <= 29 then j21529+1;
  else
    if 30 <= timej <= 44 then j23044+1;
  else
    if 45 <= timej <= 59 then j24559+1;
  else
    if 60 <= timej <= 89 then j26089+1;
  else
    if 90 <= timej < 996 then j2o90+1;

if distj = 0 then j3u1+1;
else
  if 1 <= distj <= 4 then j314+1;
else
  if 5 <= distj <= 9 then j359+1;
else
  if 10 <= distj <= 19 then j31019+1;
else
  if 20 <= distj <= 29 then j32029+1;
else
  if 30 <= distj <= 49 then j33049+1;
else
  if 50 <= distj < 996 then j3o50+1;
end;

```

```

else
if hjob = '2' and (distj ne 996 and timej ne 996) then
do;
    j2nopl+1;
    j3nopl+1;
end;

if ampm = '1' then
do;
    if ('1200' <= wtime <= '1259') or
        ('0100' <= wtime <= '0259') then j4123a+1;
    else
        if '0300' <= wtime <= '0559' then j436a+1;
    else
        if '0600' <= wtime <= '0659' then j467a+1;
    else
        if '0700' <= wtime <= '0729' then j47730a+1;
    else
        if '0730' <= wtime <= '0759' then j47308a+1;
    else
        if '0800' <= wtime <= '0829' then j48830a+1;
    else
        if '0830' <= wtime <= '0859' then j48309a+1;
    else
        if '0900' <= wtime <= '0959' then j4910a+1;
    else
        if '1000' <= wtime <= '1159' then j4104p+1;
    else
        if wtime = '' or wtime = 'D' or wtime = 'R' then
            j4nr+1;
end;
else
if ampm = '2' then
do;
    if ('1200' <= wtime <= '1259') or
        ('0100' <= wtime <= '0359') then j4104p+1;
    else
        if ('0400' <= wtime <= '1159') then j4412p+1;
    else
        if wtime = '' or wtime = 'D' or wtime = 'R' then
            j4nr+1;
end;
else
if ampm = 'D' or ampm = 'R' or ampm = '' then j4nr+1;

if (rel = 1 or rel = 2) and winus ne '1' and age >= 14 then

```

```

do;
  hhwlineq = wlineq;
  hhtran = tran;
  hhvehcl = vehcl;
  hhpass = pass;
  hhhjob = hjob;
  hhtimej = timej;
  hhdistj = distj;
  hhampm = ampm;
  hhwtime = wtime;
end;

end; /* end of wlineq check */
end; /* end of status check */
if last.ctrlnm13 then output rcd.jtwrcd;
output rcd.jtwjobs;
run;
/* ****
*      SECTION NINE
*
* Recode for lodg and ldgrs:
*
* Outputs one observation per household showing rent paid by
* nonrelative lodgers
* (Lodgers not related to the head of household and
* not having parents or spouse that are co-owners or co-renters)  */
/* ****

```

DATA rcd.rnt01rcd (KEEP = ctrlnm13 lodgr lodgrs sgrndpa sparnt);

```

length lodgr 4. lodgrs 4. sgrndpa $1 sparnt $1 sgrndpa $1 sprnln 3.;
retain lodgr 0;
retain lodgrs 0;

```

```

ARRAY pers_(30)    ;
ARRAY ldsta_(30) $1. ;
ARRAY rel_(30)    ;
ARRAY par_(30)    ;
ARRAY rntdu_(30)   ;
ARRAY ldrnt_(30)   ;
ARRAY ten_(30)    $1 ;
ARRAY age_(30)    ;
ARRAY spos_(30)   ;
ARRAY ldgnr_(30) $1 ;

```

```

/* ** Read person observations for one household group and
   load person data into arrays ***** */

persct = 0;
DO UNTIL (LAST.ctrlnm13);

  persct+1;
  merge edt.houshld (keep = ctrlnm13 status)
    edt.person;
  by ctrlnm13;

  if status = '1' then
    do;
      pers_(person) = person;
      ldsta_(person) = lodsta;
      rel_(person) = rel;
      age_(person) = age;
      par_(person) = par;
      rntdu_(person) = rntdue;
      ldrnt_(person) = lodrnt;
      spos_(person) = spos;
      ten_(person) = ten;
    end;

  END; /* DO UNTIL loop */

/* For each houshold, calculate and output household rent
   paid by nonrelatives */

if status = '1' then
  do;

    lodgrs = 0;
    lodgr = 0;

    DO i = 1 to persct;

      /* check for gen3 - COMP05 - spouse parent and spouse grandparent check */
      if rel_(i) = '1' then
        if spos_(i) > 0 then
          if par_(spos_(i)) > 0 then
            do;
              sparnt = 'X';
              sparntln = par_(spos_(i));
              if par_(sparntln) > 0 then

```

```

do;
  sgrndpa = 'X';
  put 'spouses grandparents -- ctrlnm13 = ' ctrlnm13;
end;
end; /* parent of spouse check */

/* this is the rent paid by lodger recode that has been rewritten for n99. */

if ldsta_(i) eq '1' then lodgrs +1;
if ldsta_(i) eq '1' and (rntdu_(i) ge 1 and rntdu_(i) le 52) and ldrnt_(i) ge 1 then
  lodgr + (ldrnt_(i) * rntdu_(i)) / 12;
if (i eq persct) then
  do;
    if lodgr le 0 then
      lodgr = .D ;
    end;
END; /* DO loop */

end; /* status check */

```

OUTPUT;

```

/* ** Clear arrays before next household group is read and processed ** */
lodgrs = .B;
lodgr = .B;
sparnt = '';
sgrndpa = '';
sprntln = .B;
DO i = 1 TO persct;
  pers_(i) = 0 ;
  ldsta_(i)= '';
  rel_(i)= 0 ;
  par_(i) = 0 ;
  rntdu_(i) = 0 ;
  ten_(i)= '' ;
  ldrnt_(i)= 0 ;
  age_(i)= 0 ;
  spos_(i)= 0 ;
  ldgnr_(i) = '';
END;

```

RUN;

```

*****  

*      SECTION TEN

```

```
*  
*   Merge all previous data sets together and do the rounding on the  
*   calculated variables  
******/
```

```
data rcdtblone;
```

```
merge rcd.perrcd
```

```
    rcd.hhrbcd
```

```
    rcd.povrcd
```

```
    rcd.inc3rcd
```

```
    rcd.mtgrcd
```

```
    rcd.mtg2rcd
```

```
    rcd.mv2rcd
```

```
    rcd.jtwrcd
```

```
    rcd.rnt01rcd;
```

```
by ctrlnm13;
```

```
confeetr = round(confeetr, 1);
```

```
zcrowfr = round(zcrowfr, 1);
```

```
zcrowdr = round(zcrowdr, .01);
```

```
zcrowbr = round(zcrowbr, .01);
```

```
zsmhcnr = round(zsmhcnr,1);
```

```
zsmhcmr = round(zsmhcmr,1);
```

```
zsmhcov = round(zsmhcov,1);
```

```
zsmhcpr = round(zsmhcpr,1);
```

```
zvir = round(zvir, .01);
```

```
amtomo = round(amtomo,1);
```

```
amtimo = round(amtimo,1);
```

```
amtwmor = round(amtwmor,1);
```

```
amttmor = round(amttmor,1);
```

```
amtgmor = round(amtgmor,1);
```

```
amtfmor = round(amtfmor,1);
```

```
lrntmor = round(lrntmor,1);
```

```
amttxpr = round(amttxpr,1);
```

```
amttxvr = round(amttxvr,1);
```

```
mainr = round(mainr,1);
```

```
poorr = round(poorr, .01);
```

```
otpnr = round(otpnr,1);
```

```
clpevar = round(clpevar,1);
```

```
crinrr = round(crinrr,.1);
```

```
prin01 = round(prin01r,1);
```

```
lodgr = round(lodgr,1);
```

```
run;
```

```

/* ****
*      SECTION ELEVEN
*
* Final section --- reset rounded variables to special missings if they
*                  were before rounding and calculate gen3 used in the
*                  COMP05 table stub.
* ****
data rcd.tblrcd ;

length gen3 $1;

set rcdtblone;

if confeet = . then
  confeet = confeetr;

if zcrowf = . then
  zcrowf = zcrowfr;

if zcrowd = . then
  zcrowd = zcrowdr;

if zcrowb = . then
  zcrowb = zcrowbr;

if zsmhcn = . then
  zsmhcn = zsmhcnr;

if zsmhcm = . then
  zsmhcm = zsmhcml;

if zsmhco = . then
  zsmhco = zsmhcpr;

if zsmhcv = . then
  zsmhcv = zsmhcpr;

if zsmhcp = . then
  zsmhcp = zsmhcpr;

if zvi = . then
  zvi = zvir;

if amtomo = . then

```

```
amtomo = amtomor;  
if amtomo = 0 then  
    amtomo = 1;
```

```
if amtimo = . then  
    amtimo = amtimor;  
if amtimo = 0 then  
    amtimo = 1;
```

```
if amtwmo = . then  
    amtwmo = amtwmor;  
if amtwmo = 0 then  
    amtwmo = 1;
```

```
if amttmo = . then  
    amttmo = amttmor;  
if amttmo = 0 then  
    amttmo = 1;
```

```
if amtgmo = . then  
    amtgmo = amtgmor;  
if amtgmo = 0 then  
    amtgmo = 1;
```

```
if amtfmo = . then  
    amtfmo = amtfmor;  
if amtfmo = 0 then  
    amtfmo = 1;
```

```
if lrntmo = . then  
    lrntmo = lrntmor;
```

```
if amtxp = . then  
    amtxp = amtxpr;
```

```
if amtxv = . then  
    amtxv = amtxvr;
```

```
if maint = . then  
    maint = maintr;
```

```
if poor = . then  
    poor = poorr;
```

```
if otpin = . then  
    otpin = otpinr;
```

```
if clpeva = . then
    clpeva = clpevar;

if crintr = . then
    crintr = crinrr;

if prin01 = . then
    prin01 = prin01r;

if lodg = . then
    lodg = lodgr;

/* gen3 recode -- COMP05 -- set gen3 if one of three possibilities exist */

if (parent = 'X' or sparnt = 'X') and hhldkid > 0 and refper = 'X' then
    gen3 = 'X';

if refper = 'X' and hhldkid > 0 and grankid = 'X' then
    gen3 = 'X';

if refper = 'X' and (parent = 'X' or sparnt = 'X') and (grandpa = 'X'
or sgrndpa = 'X') then
    gen3 = 'X';

run;
/* end of table recode program */
```